Fundamentos de Linux

Félix Molina Ángel

Fundamentos de Linux

Objetivo: Identificar los componentes físicos y lógicos con los que interactúa Linux, para facilitar la interpretación y ejecución de comandos de usuario Linux.

Dirigido a: Usuarios nuevos en Linux y Unix, Usuarios y Administradores de otros OS.

-Experiencia a nivel de usuario en el manejo del teclado, mouse, menús y cualquier GUI de cualquiee sistema de cómputo.

Contenido

- Introducción
- Iniciación en Linux
- Comandos básicos y ayuda
- Navegando en el sistema de archivos
- Usuarios, Grupos y Permisos
- Introducción al bash shell
- E/S estandar y pipes
- Herramientas de procesamiento de texto
- Editor de Textos vim
- Administración de procesos

Introducción

- Objetivos:
  - Explicar la naturaleza de los programas open source.
  - Listar algunas distribuciones Linux y su alcance.
  - Explicar las características principales de Linux.

Introdución

- Qué es Open Source
  - Código fuentes y aplicaciones disponibles para todos.
  - Libertad para distribuir el software y código fuente
  - Integridad del código del autor.
- Software libre da libertad para:
  - Ejecutar el programa
  - Estudiar como trabaja un programa, y modificarlo para hacer que haga lo que queremos.
  - Distribuir copias del programa para ayudar a otras personas.
  - Distribuir copias de tus versiones modificadas a otros.

Introducción

- Como surge Linux
  - En 1984, El proyecto GNU y la Fundación de Software Libre (FSF).
    - Crean una versión open source de las utilidades de Unix.
    - Crean la Licencia Pública General (GPL), que forza los principios de open source.
  - En 1991, Linus Torvalds crea un kernel tipo Unix de open source, liberado bajo GPL.
  - Hoy, se cuenta con Kernel Linux más uitlidades, completamente open source.

Introducción

- Linux o Unix?
- Características de operación
  - Multiusuario
  - Multitarea
  - Multiproceso
- Distribuciones
  - De Escritorio
  - De Servidor
- Ambientes de Operación
  - Shell
  - XWindows

Introducción

- Principios de Linux
  - Cualquier cosa en Linux, se trata como un archivo
  - Los programas de propósito simple.
  - Preparado para encadenar programas a fin de realizar tareas complejas.
  - Evita interfaces de usuario cautivas.
  - Las configuraciones se almacenan en archivos de texto.

Introducción

- Formas de Conexión
  - Por Consola
  - Desde una terminal tonta
  - Desde una terminal virtual a través de la red
- Tipo de Usuarios
  - Usuario root
  - Usuarios especiales
  - Usuarios normales
- Grupos de Usuarios
  - De sistema
  - De Servicios o aplicaciones
  - Normales
- Establecimiento de una sesión Linux
  - Prompt de Inicio
  - Cambio de contraseña

Iniciación en Linux

- Al terminar esta unidad, debes ser capaz de:
  - Ingresar a un sistema Linux
  - Iniciar X desde una consola
  - Accesar a la línea de comandos desde X
  - Cambiar tu password
  - Entender la naturaleza de los privilegios de root
  - Promover tus privilegios
  - Editar archivos de texto

Inciación en Linux

- Instalación de Vector Linux en una VM
  - Seguir las instrucciones que indica el Wizard de Instalación.
  - Tiempo estimado: 15 Minutos

Iniciación en Linux

- Entrada a un sistema Linux
  - Hay dos tipos de pantalla: Consolas virtuales e interfaces gráficas.
  - Registro utilizando un nombre de login y una contraseña
  - Cada usuario tiene un directorio home para almacenar sus archivos personales.

Iniciación en Linux

- Cambiando entre las consolas virtuales y el ambiente gráfico.
- Un sistema Linux común ejecutará seis terminales virtuales y una consola gráfica.
- Cambiar de una consola a otra pulsando [Ctrl Alt F1] … [CTRL Alt F6].
- Entra a la consola gráfica con [Ctrl Alt F7].

Iniciación en Linux

- Componentes de un sistema X Windows
  - X Windows es un subsistema gráfico de Linux.
  - Xorg es una versión particular de X, utilizada por varios sistemas Linux.
  - La apariencia y comportamiento es ampliamente controlado por el ambiente de escritorio.
  - En Linux existen más de un ambiente de escritorio, Gnome y KDE son los más conocidos, pero no los únicos.

Iniciación en Linux

- Arranque del servidor X
  - En algunos sistemas, el servidor X inicia automáticamente en tiempo de arranque.
  - De lo contrario, si el sistema inicia en consola virtual, los usuarios deben arrancar el servidor X manualmente.
    - El servidor X debe ser pre-configurado por el administrador del sistema.
    - Entra a una consola virtual y ejecutar startx
    - El servidor aparece en [Ctrl Alt F7]

Iniciación en Linux

- Cambio de contraseña
  - Cambia tu contraseña la primera vez que entres.
  - Selecciona una contraseña que sea dificil de adivinar.
  - Cámbiala cada cierto tiempo.
  - Para cambiar tu contraseña en KDE, Selecciona Inicio->Settings->Security & Privacy ->Password & Account
  - Desde una terminal, ejecuta **passwd**

Iniciación en Linux

- El usuario root es una cuenta administrativa especial.
  - También se conoce como superusuario o administrador.
  - root tiene el control total del sistema.
    - Puede hacer y destruir cualquier parte del sistema.
  - No ingreses como **root**, a menos que sea necesario.
  - Los usuarios normales (no privilegiados) tienen capacidades limitadas.

Iniciación en Linux

- Cambio de Identidad
  - su – crea un nuevo shell para root.
  - sudo *comando* ejecuta un comando que sólo root puede ejecutar.
    - Requiere configuración previa por el administrador del sistema.
  - id Muestra información del usuario actual.
  - su – username Se cambia al shell del usuario especificado.

Edición de archivos de texto

- El editor nano
  - Fácil de aprender y fácil de utilizar.
  - No tan rico en características como otros editores avanzados.
- Otros editores
  - gedit, un editor gráfico.
  - vim, un editor avanzado
  - gvim, la versión gráfica de vim

Comandos básicos y ayuda

- Al terminar esta unidad, debes ser capaz de:
  - Ejecutar comandos
  - Explicar el propósito y uso de algunos comandos básicos.
  - Utilizar los recursos de ayuda interconstruída.

Ejecución de comandos

- Ejecución de comandos
  - Los comandos se escriben con la siguiente sintaxis.
    - **comando** *opciones* *argumentos*
  - Los elementos de un comando se separan por espacios.
  - Las opciones modifican el comportamiento por default de un comando.
  - Las opciones de una simple letra son precedidas por -
    - Y pueden escribirse como -a -b -c o bien como -abc
  - Las opciones de palabra son precedidas generalmente por --
    - Por ejemplo: --help
  - Los argumentos son archivos u otros datos requeridos por el comando
  - Se pueden ejecutar múltples comandos por línea, separandolos por ;
  - Algunos comandos simples
    - date
    - cal

Ejecución de comandos

- Obtención de ayuda
  - No es necesario memorizar todo
  - Linux proporciona varios niveles de ayuda
    - **whatis**
    - **comando** –-help
    - **man** comando
    - **info** comando
  - Los manuales de ayuda están bajo /usr/share/man
  - Documentación en línea

Ejecución de comandos

- El comando whatis
  - Despliega descripciones cortas de los comandos
  - Utiliza una base de datos que se utiliza diariamente
  - Es posible que no este disponible inmediatamente después de instalar.
  - $ **whatis** date - despliega o configura la fecha y hora del sistema.

Ejecución de comandos

- La opción --help
  - Despliega un resúmen de uso y lista de argumentos.
  - Utilizada por la mayoría de comandos

$date –help

  - Usage: date [OPTION]... [+FORMAT] or:
  - date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]
  - Display the current time in the given FORMAT,
  - or set the system date.
  - ...argument list omitted...

Ejecución de comandos

- Leyendo resúmenes de utilización.
  - Desplegados por –help, man y otros.
  - Utilizados para describir la sintaxis de un comando
    - Los argumentos entre [ ] son opcionales.
    - Los argumentos en mayúsculas o <> son variables.
    - El texto seguido por … representa una lista
    - x|y|a significa “x o y o z”
    - -abc significa cualquier combinación de -a, -b y -c.

Ejecución de comandos

- El comando man
  - Proporciona documentación para comandos.
  - Casi cada comando tiene una página man.
  - Las páginas se agrupan en capítulos.
  - Colectivamente son referidas como el manual Linux.
  - man [<capítulo>] <comando>

Ejecución de comandos

- Navegando en las páginas man
  - Mientras ves una página man
    - Navega con las flechas, PgUp, PgDn
    - /texto para buscar el texto especificado
    - n/N Va a la siguiente o a la ocurrencia anterior.
    - q Sale de la ayuda.
  - Buscando el manual
    - man -k palabra clave Lista todas las páginas que coincidan con la palabra clave.
    - Utiliza la base de datos whatis

Ejecución de comandos

- El comando info
  - Es similar a man, pero a menudo más detallado.
  - Ejecuta **info** sin argumentos para listar todas las páginas.
  - Las páginas info, se estructuran como un sitio Web
    - Cada página se divide en nodos.
    - Los enlaces a nodos son precedidos por un *
    - info [comando]

Ejecución de comandos

- Navegando en páginas **info**
  - Mientras ves una página info
    - Navega con las flechas, **PgUp**, **PgDn**
    - **Tab** te mueve al siguient enlace.
    - **Enter** sigue el enlace seleccionado
    - **n/p/u** Va al siguiente, previo, o un nodo arriba.
    - s texto Busca el texto especificado.
    - **q** sale de info.

Ejecución de comandos

- Documentación extendida
  - El directorio /usr/share/doc
    - Subdirectorios para la mayoría de paquetes instalados
    - Lugar para los documentos que no encajan en otro lugar.
    - Ejemplos de archivos de configuración
    - Documentación html/pdf/ps
    - Detalles de licenciamiento

Revisando el sistema de archivos

- Al terminar esta unidad, debes ser capaz de:
  - Describir los elementos importantes de un filesystem.
  - Copiar, mover y eliminar archivos
  - Crear y ver archivos.

Revisando el sistema de archivos

- Conceptos de jerarquía de archivos
  - Los archivos y directorios son organizados en una estructura de árbol invertido.
  - El sistema de archivos (filesystem), inicia en el directorio raíz (root) representado por una /
  - Los nombres son sensibles a mayúsculas y minúsculas.
  - Las rutas son delimitadas por /

Revisando el sistema de archivos

- Algunos directorios importantes
  - **Directorios home**: /root/, /home/username
  - **Ejecutables de usuario**: /bin, /usr/bin, /usr/local/bin
  - **Ejecutables del sistema**: /sbin, /usr/sbin, /usr/local/sbin
  - **Otros puntos de montaje**: /media, /mnt
  - **Configuraciones**: /etc
  - **Archivos temporales**: /tmp
  - **Kernels y cargador de arranque**: /boot
  - **Información del sistema**: /proc, /sys
  - **Librerías compartidas**: /lib, /usr/lib, /usr/local/lib

Revisando el sistema de archivos

- Directorio de trabajo actual
  - Cada shell y proceso de sistema tiene un directorio de trabajo actual (cwd)
  - **pwd**
    - Despliega la ruta absoluta para el cwd del shell.

Navegando por el sistema de archivos

- Nombres de archivos y directorios
  - Los nombres pueden ser de hasta 255 caracteres
  - Todos los caracteres son válidos, excepto la /
    - No se recomienda utilizar ciertos caracteres especiales.
    - Algunos caracteres deben protegerse por “ al referenciarlos.
  - Los nombres son sensibles a mayúsculas y minúsculas.
    - Por ejemplo: CORREO, Correo, correo

Navegando por el sistema de archivos

- Rutas absolutas y relativas
  - Rutas absolutas
    - Inician con una /
    - Mapeo de ruta completa para ubicar un archivo
    - Puede utilizarse para especificar el nombre de un archivo
  - Rutas relativas
    - No inician con /
    - Especifican una ubicación relativa al directorio de trabajo actual.
    - Puede utilizarse como una forma más corta de especificar el nombre de un archivo

Navegando por el sistema de archivos

- Cambiando entre directorios
  - **cd** cambia de directorio
    - Dando una ruta absoluta o relativa
      - **cd** /usr/local
      - **cd** bin
    - Moviendose al directorio superior (padre)
      - **cd** ..
    - Ir al directorio home
      - **cd**
    - Ir al directorio de trabajo anterior
      - **cd -**

Navegando por el sistema de archivos

- Listando el contenido de directorios
  - Despliega el contenido del directorio actual o de un directorio especificado
    - **ls** [opciones] [archivos o directorios]
  - Ejemplos
    - **ls** *Lista los archivos del directorio actual*
    - **ls** -a *La lista incluye los archivos ocultos*.
    - **ls** -l *Lista un archivo por línea con información detallada*
    - **ls** -R *Devuelve una lista recursiva*

Navegando por el sistema de archivos

- **cp** Para copiar archivos y directorios
  - **cp** [opciones] archivo destino
- Más de un archivo pueden copiarse de una sóla vez, si destino es un directorio
  - **cp** [opciones] archivo1 archivo2 ... destino
- Si el destino es un archivo, la copia sobreescribe el archivo destino, en caso de que exista.
- Si el destino es un directorio, una copia de los archivos se colocan en éste.
- Si el destino no existe, la copia se graba con el nombre de destino.

Navegando por el sistema de archivos

- **Ejemplos con cp**

cp /etc/hosts /tmp

  – Copia el archivo hosts desde /etc al directorio /tmp

cd /tmp Se cambia a /tmp

cp hosts hosts1 hace una copia de hosts y le pone nombre hosts1

cp -f /etc/rc.d/rc.inet1.conf . La opción -f sobreescribe el archivo en el destino, si existe

cp -r . /opt/practicas Copia el directorio actual de manera recursiva, en /opt/practicas.

Navegando por el sistema de archivos

- **mv** Para mover y renombrar archivos y directorios
  - mv [opciones] objeto destino
  - Si destino no existe, objeto cambia al nombre de destino
  - Si destino es un directorio, el objeto se mueve a ese directorio.
- Si se especifica más de un objeto, el destino debe ser un directorio.
  - mv [opciones] objeto1 objeto2 ... destino
- Ejemplos:
  - $ echo “Este texto originalmente se crea en el home directory” >$HOME/aroriginal
  - $ echo “

Navegando por el sistema de archivos

- Creación y eliminación de archivos
  - **touch** Crea archivos vacíos o actualiza su timestamp.
    - **touch** [opciones] archivo
  - **rm** Elimina archivos
  - **rm** [opciones] <archivo1> ...
  - Ejemplos:
  - **rm** -i archivo1 (es interactivo)
  - **rm** -f archivo1 (Forza la eliminación)
  - **rm** -r directorio1 (Hace un borrado recursivo)

Navegando por el sistema de archivos

- Creación y eliminación de directorios
  - Creación de directorios
    - **mkdir** [opciones] directorio1 directorio2 …
  - Eliminación de directorios vacíos
    - **rmdir** directorio1 directorio2 …
  - Elminación recursiva de un árbol de directorios
    - **rm** -r directorio

Navegando por el sistema de archivos

- Determinando el contenido de un archivo
  - Los archivos pueden contener diferentes tipos de datos.
  - Es muy útil revisar el tipo de archivo antes de abrirlo, para determinar el comando o aplicación a usar para accesarlo.
  - **file** [opciones] <archivo1> <archivo2> ...

Usuarios, Grupos y Permisos

- Objetivos: Al terminar esta unidad, debes ser capaz de:
  - Explicar el modelo de seguridad de Linux
  - Explicar el propósito de las cuentas de usuario y grupo.
  - Leer y configurar permisos de archivos

Usuarios, Grupos y Permisos

- **Usuarios**
  - A cada usuario se le asigna un ID de usuario único (UID)
    - El UID 0 identifica a root
  - Los nombres de usuario y UID's se almacenan en /etc/passwd
  - A cada usuario se le asigna un directorio home y un programa que se ejecuta cuando entra al sistema (generalmente un shell).
  - Los usuarios no pueden leer, escribir o ejecutar archivos de otros, sin permisos.

Usuarios, Grupos y Permisos

- **Grupos**
  - Los usuarios son asignados a grupos
  - A cada grupo se le asigna un ID de grupo único (GID)
  - Los GID se almacenan en /etc/group
  - A cada usuario se le asigna un grupo privado.
    - Puede ser agregado a otros grupos para acceso adicional.
  - Todos los usuarios de un grupo pueden compartir archivos que pertenecen al grupo.

Usuarios, Grupos y Permisos

- **Seguridad de archivos Linux**
  - Cada archivo es propiedad de un UID y un GID.
  - Cada proceso se ejecuta como un UID y uno o más GID.
    - Generalmente determinado por quién ejecuta el proceso.
  - **Tres categorías de acceso**
    - Procesos ejecutandose con el mismo UID que el archivo (user)
    - Procesos ejecutándose con mismo GID del archivo (grupo)
    - Otros procesos (otros)

Usuarios, Grupos y Permisos

- **Precedencia de permisos**
  - Si coincide el UID, se aplican los permisos de usuario.
  - De otra manera, si coincide el GID, se aplican los permisos de grupo.
  - Si ninguno coincide, se aplican los permisos de otros.

Usuarios, Grupos y Permisos

- **Tipos de permisos**
  - Se utilizan cuatro símbolos al desplegar los permisos.
    - r: Permiso para leer el contenido de un archivo o listar el contenido de un directorio.
    - w: Permiso para modificar el contenido de un archivo o crear y eliminar archivos en un directorio.
    - x: Permiso para ejecutar un programa o cambiarse a un directorio.
    - -: Sin permiso (en lugar de la r,w, o x)

Usuarios, Grupos y Permisos

- **Examinando e interpretando los permisos**
  - Los permisos pueden visualizarse con ls -l
  - $ ls -l

```
drwxr-xr-x 4 molina molina      128 2010-06-22 10:49 pt/
-rw-r--r-- 1 molina molina    38668 2010-09-23 23:38 npatgpc.dl_
-rw-r--r-- 1 molina molina  1473551 2010-12-13 12:12 oficiosre.pdf
-rw-r--r-- 1 molina molina   281600 2010-08-30 13:02 plandetrabajo.doc
```

  - El tipo de archivo y permisos se representan con una cadena de 10 caracteres.

Usuarios, Grupos y Permisos

- Cambiando la posesión de un archivo
  - Sólo root puede cambiar al propietario de un archivo.
  - Sólo root o el propietario puede cambiar el grupo de un archivo.
  - La posesión es cambiada con chown
    - **chown** [-R] username archivo | directorio
  - La posesión de grupo se cambia con chgrp
    - **chgrp** [-R] grupo archivo | directorio

Usuarios, Grupos y Permisos

- **Cambiando permisos**: Método simbólico
  - Para cambiar los modos de acceso se utiliza
    - **chmod** [-R] modo archivo
  - Donde modo es:
    - **u**,**g**, u **o** para usuario, grupo u otro.
    - a Todos los usuarios.
    - **+** o **–** para otorgar o negar.
    - **r**, **w** o **x** para leer, escribir o ejecutar
  - Por ejemplo:
    - ugo+r Otorga lectura para todos
    - También puede escribirse como a+r
    - o-wx Quita los permisos de escritura y ejecución a otros

Usuarios, Grupos y Permisos

- Cambiando permisos: Método númerico
  - Utiliza un modo númerico de tres digitos.
    - El primer dígito especifica los permisos del propietario.
    - El segundo dígito especifica los permisos del grupo.
    - El tercer dígito especifca los permiso para otros
  - Los permisos se calculan agregando:
    - 4 Para lectura
    - 2 Para escritura
    - 1 Para ejecución
  - Ejemplo:
    - chmod 640 mydoc

Utilizando el shell bash

- Objetivos: Al terminar la unidad, debes ser capaz de:
  - Utilizar atajos de línea de comandos.
  - Expandir la línea de comandos.
  - Utilizar trucos de edición y history.

Utilizando el shell bash

- Atajos de línea de comandos, englobar archivos
  - Englobar es expansión de comodines
    - *, coincide con cero o más caracteres
    - ?, coincide con cualquier simple caracter.
    - [0-9] coincide con un rango de números.
    - [abc] coinde con cualquiera de los caracteres de la lista.
    - [^abc] coincide con todo, excepto con los caracteres de la lista.

Utilizando el shell bash

- Atajos de la línea de comandos: la tecla TAB
  - Pulsa **Tab** para completar líneas de comando.
    - Para un nombre de comando, completará su nombre.
    - Para un argumento, completará el nombre del archivo.
  - Ejemplos:
  - $ xte<tab>
  - $ xterm
  - $ ls my<tab>
  - $ ls mydocumento

Utilizando el shell bash

- Atajos de línea de comandos: History
  - Bash almacena un historial de comandos que has introducido, los cuales pueden utilizarse para repetir comandos.
  - Utiliza el comando history para ver la lista de comandos recordados
  - $ history

Utilizando el shell bash

- Expansion de la línea de comandos: la tilde (~)
  - Puede referirse al directorio home.
  - $ cat ~/.bash_profile
  - O puedes referirte al directorio home de otro usuario.
  - $ cat ~dalia/.bash_profile

Utilizando el shell bash

- Expansión de línea de comandos: Comandos y { }
  - Expansión de comandos: $( ) o ` `
    - Imprime la salida de un comando como argumento para otro comando.

$ echo “El nombre del sistema es: $(hostname)”

  - Expandiendo con: { }
    - Abreviación para escribir cadenas repetitivas:
      - $echo curso{1,2,3}

curso1 curso2 curso3

Utilizando el shell bash

- **Lo básico de los scripts**
  - Los shells scripts son archivos de texto que contienen una serie de comandos o instrucciones para ejecutarse.
  - Los shell scripts son útiles para
    - Automatizar comandos comúnmente utilizados.
    - Ejecutar administración de sistemas y resolución de problemas.
    - Crear aplicaciones simples.
    - Manipular texto o archivos.

Utilizando el bash shell

- Paso1: Creando shell scripts
  - Utiliza un editor tal como vi para crear un archivo de texto que contenga comandos.
  - La primera línea contiene la secuencia mágica: #!/bin/bash Se utiliza para indicar donde se encuentra el shell a utilizar.
  - Documenta tus script, coloando comentarios con #
    - #Este es un comentario

Creando shell scripts

- Paso 2: Crea es script ejecutable
  - $ chmod u+x miscript.sh
- Para ejecutar el nuevo script
  - Coloca el archivo de script en un directorio de la ruta de ejecutables.
  - O bien, especifica la ruta absoluta o relativa al script, en la línea de comandos.

Creando shell scripts

- Ejemplo de un shell script

```
#!/bin/bash
echo “Bienvenido $USER”
echo “al sistema $(hostname)”
echo “Hoy es: `date`”
echo “tu directorio home es: $HOME”
```

Ls -a

E/S estandar y Pipes

- Al completar esta unidad, debes ser capaz de:
  - Redirigir los canales de E/S hacia archivos
  - Conectar comandos utilizando pipes
  - Utilizar el ciclo for para iterar sobre conjuntos de valores.

E/S Estandar y pipes

- Entrada y salida estandar
  - Linux proporciona tres canales de E/S para los programas.
  - Entrada estandar (STDIN) - teclado por default
  - Salida estandar (STOUT) - ventana de terminal por default.
  - Error estandar (STDERR) – Ventana de terminal por default.

E/S Estandar y pipes

- Redireccionando la salida a un archivo.
  - STDOUT y STDERR pueden ser redireccionados a un archivo.
    - Comando operador archivo
  - Los operadores soportados son:
    - > redirige STDOUT a un archivo
    - 2> redirige STERR a un archivo
    - &> redirige toda la salida a un archivo
  - El contenido del archivo se sobrescribe por default.
  - Se utiliza >> para agregar.

E/S Estandar y pipes

- Ejemplos de redirección de salida a un archivo
  - Este comando genera salida y errores cuando se ejecuta como no-root.
    - $ find / -name hosts
  - Se pueden utilizar los operadores para almacenar la salida y los errores.
    - $ find / -name hosts > salida
    - $ find / -name hosts 2>errores
    - $ find / -name hosts >salida.out 2>errores

cisco
Networking Academy

E/S Estandar y pipes

- Redirigiendo STDOUT a un programa (piping)
- El caracter | puede conectar comandos.
  - Comando1 | comando2
  - Envía STDOUT del comando1, como STDIN del comando2, en lugar de la pantalla.
  - STDERR no es reeviado a través de pipes.
  - Utilizado para combinar la funcionalidad de múltiples herramientas.
  - comando1|comando2|comando3

E/S Estandar y pipes

- Redirección de E/S: Ejemplos
  - Este comando genera salida y errores cuando se ejecuta como no-root.

$ find /etc -name passwd

  - Para evitar esta situación pueden aplicarse los operadores, a fin de redireccionar la salida y los errores.
  - $ find /etc -name passwd >/tmp/salidapass.txt
  - $ find /etc -name passwd 2>/tmp/errorespass.txt
  - $ find /etc -name passwd >/tmp/salidapass.txt 2>/tmp/errores

E/S Estandar y pipes

- Redireccionando STDOUT a un programa (piping)
  - El caracter | se utiliza para conectar comandos

Comando1 | comando2

  - Envía STDOUT del comando 1 como STDIN del comando2.
  - STDERR no reenvía a través de pipe.
- Las pipes son utilizadas para combinar la funcionalidad de múltiples herramientas.
  - Comando1 | comando2 | comando3 ...

E/S Estandar y pipes

- Redirigiendo STDOUT a un programa.
  - Eneste ejemplo, less toma como entrada, la salida de ls -l y la despliega página por página.
  - $ ls -l /usr/bin | less
  - cat /etc/inet1.conf | less

E/S Estandar y pipes

- Redireccionando STDIN desde un archivo.
  - La entrada estandar se redirige con <.
  - Algunos comandos pueden aceptar datos redirigidos a STDIN desde un archivo.
  - $ tr 'A-Z' 'a-z' < .bashrc
  - Este comando convierte las letras mayusculas del contenido de .bashrc y lo envía a STDOUT.
  - $ tr 'a-z' 'A-Z' < .bashrc

E/S Estandar y pipes

- **Scripts: Ciclos for**

```
CURSO="Fundamentos de Linux"
for MAESTRO in juan jorge elena
do
echo "Profesor: $MAESTRO"
echo "Presente:"
echo "A traves de este medio, me permito informar"
echo "a usted, que el curso de $CURSO, sera"
echo "impartido a partir de hoy `date` a las 9:00"
echo -e "\n\n "
echo "Atentamente"
echo "Depto. De Desarrollo Academico"
done
```

Herramientas de procesamiento de textos

- Al terminar esta unidad serás capaz de:
  - Utilizar herramientas para extraer y manipular archivos de texto.

Herramientas de procesamiento de textos

- Herramientas para extraer texto
  - Revisar el contenido de un archivo: **less**, **more** y **cat**.
  - Extraer un fragmento de un archivo: **head**, **tail**.
  - Extraer por columnas de un archivo: **cut**
  - Extraer por palabras clave: **grep**

Herramientas de procesamiento de textos

- Ver el contenido de un archivo: cat, more, less
  - **cat** envia uno o más archivos a STDOUT
  - Múltiples archivos pueden ser concatenados.
  - **more** muestra el contenido de un archivo de texto, página por página.
    - Se pulsa la barra para continuar o q para salir.

Herramientas de procesamiento de textos

- **less** muestra un archivo en STDOUT, una página a la vez, con las siguientes facilidades.
  - Permite utilizar las teclas **PgDn** o **PgUp** para mover una página adelante o hacia atrás.
  - **/texto** Para buscar una ocurrencia del **texto** especificado.
  - **n**/**N** Para encontrar la siguiente o la anterior ocurrencia del texto especificado.
- **less** es el paginador utilizado por **man**

Herramientas de procesamiento de textos

- Extracción de fragmentos de un archivo: **head** y **tail**
  - **head** despliega las primeras 10 líneas del archivo especificado.
    - Utiliza -n para especificar las líneas a desplegar.
  - **tail** despliega las últimas 10 líneas del archivo especificado.
    - También puede utilizarse -n para especificar la cantidad de líneas a desplegar.
    - **tail** es muy útil para monitorear archivos de log

Herramientas de procesamiento de textos

- Extracción de texto por palabra clave: **grep**
  - Imprime cada linea del archivo o de STDIN, en la que existe una coincidencia con el patrón especificado.
  - Utiliza la opción **-i** para no diferenciar entre mayúsculas y minúsculas.
  - Utiliza **-v** para desplegar las líneas que no contienen el patrón.

Herramientas de procesamiento de textos

- Extracción de texto por columna: **cut**
  - Despliega columnas específicas o de STDIN.
  - Utiliza **-d** para especificar el delimitador de columna. El default es TAB
  - Utiliza **-f** para especificar las columnas a imprimir, separandolas por (**,**).
  - Utiliza **-c** para cortar por caracteres.

Herramientas de procesamiento de textos

- Ejemplos de extracción, con **more**, **less** y **cat**

$ ls -l | more

$ more /etc/rc.d/rc.inet1.conf

$ less /etc/rc.d/rc.inet1.conf

$ cat /etc/group

Herramientas de procesamiento de textos

- Ejemplos de extracción de texto con head, tail, grep y cut.
  - $ man cp | cat | head
  - $ head /var/log/

  - $ tail programas.txt
  - $ grep virus unvirus.txt
  - $cut -d: -f1,4 /etc/passwd

Editor de Textos vim

- Objetivo: Al terminar esta unidad, debes ser capaz de:
  - Utilizar los tres principales modos de vi
  - Navegar a través del texto y entrar en modo de inserción.
  - Cambiar y borrar texto
  - Grabar y salir.

Editor de Textos vim

- Introducción a vim
  - Es la nueva versión de vi, el editor de textos estandar de Unix.
  - Al ejecutar vi, corre vim por default.
  - Ventajas:
    - Es rápido: se hace más con menos keystrokes.
    - Simplicidad: no depende del mouse ni GUI.
    - Disponibilidad: Incluído con la mayoría de OS tipo Unix.

Editor de Textos vim

- **vim**: Un editor modal
  - El comportamiento de las teclas depende de los modos de **vim**.
  - Tres modos principales:
    - Modo de comandos(default): Mueve el cursor, corta y pega texto, cambia el modo.
    - Modo de inserción: Modifica texto
    - Modo de salida: Grabar, salir, etc.
  - Esc: Sale del modo actual
  - Esc Esc: Siempre regresa al modo de comando.

Editor de Textos vim

- Lo esencial de vim
  - Para utilizar vim, debes ser capaz de:
    - Abrir un archivo
    - Modificar un archivo (modo de inserción)
    - Grabar a un archivo (modo ex)

Editor de Textos vim

- Abrir un archivo en **vim**
  - Para iniciar **vi,** ejecuta:
    - **vim** filename
    - Si el archivo existe, el archivo es abierto y el contenido es desplegado.
    - Si el archivo no existe, **vi** lo crea cuando lo editado es grabado la primera vez.

Editor de Textos vim

- Modificando un archivo: Modo de Inserción
  - **i** inicia el modo de inserción en la posición del cursor.
  - Otras opciones de inserción son:
    - a Agrega después de la posición del cursor.
    - A Agrega al final de la línea actual.
    - I Inserta al principio de la línea actual
    - o Inserta una nueva línea bajo la posición del cursor.
    - O Inserta una nueva línea por arriba de la posición del cursor

Editor de Textos vim

- Grabar un archivo y salir de vim: Modo Ex
  - Entra en modo Ex con (**:**)
    - Crea un prompt de comandos en la parte inferior izquierda de la pantalla.
  - Comandos de escribir/salir comunes
    - :w Escribe (salva) el archivo a disco.
    - :wq Escribe y sale.
    - :q! Sale aunque los cambios se pierdan.
    - :q Salida normal de **vim**

Administración de Procesos

- Objetivo: Al terminar esta unidad, debes ser capaz de:
  - Explicar que es un proceso.
  - Describir como administrar procesos
  - Utilizar herramientas de control de trabajos

Administración de Procesos

- Qué es un proceso?
  - Es un conjunto de instrucciones que se cargan y ejecutan en la memoria.
  - Utilizan un ID de proceso para su identificación
  - El UID y GID determinan el acceso del filesystem
    - Normalmente heredados del usuario que lo ejecuta.

Administración de Procesos

- Listado de procesos
  - Vista de información de procesos con **ps**
    - Muestra los procesos de la terminal actual por default.
    - -a Incluye los procesos en todas las terminales.
    - -x Incluye procesos no ligados a terminales.
    - -u incluye información del propietario del proceso
    - -f Imprime el parentesco de los procesos.

Administración de Procesos

- Señales
  - La comunicación fundamental inter-procesos
  - Se envían directamente a los procesos, sin requerir una interface de usuario.
  - Los programas asocian acciones con cada señal.
  - Las señales se especifican por nombre o por número, al enviarse
    - 15 (TERM) Termina de modo normal.
    - 9 (KILL) Lo termina inmediatamente.
    - 1 (HUP) Relee los archivos de configuración

Administración de Procesos

- Enviando señales a los procesos
  - Por PID: kill [señal] pid …
  - Por nombre: killall [señal] com
  - Por patrón: pkill [-señal] pattern

Administración de Procesos

- Control de trabajos
  - Para ejecutar un proceso en background
    - Agrega un & al final de la línea de comandos: firefox &
    - Temporalmente detiene un programa ejecutandose.
      - Utiliza **Ctrl z** o bien envía la señal 17 (STOP)
    - Administra los trabajos suspendidos o en background.
    - Lista los números de trabajo y nombres: jobs
    - Resume en el background: bg
    - Resume en fore ground; fg
    - Envia una señal: kill -señal %númerodetrabajo

Administración de procesos

- Localización de procesos.
  - La forma más flexible es:
    - $ ps opciones | otros comandos
  - Por ejemplo:
    - $ ps Lista los procesos del usuario actual.
    - $ ps axl | grep molina
    - $ ps ax
    - $ ps

Administración de procesos

- Señales
  - Proporcionan una comunicación fundamental entre procesos.
  - Se envían directamente a los procesos, sin una interface requerida.
  - Los programas ejecutan acciones, de acuerdo a la señal.
  - Las señales se pueden especificar por nombre o por número, al ser enviadas.
    - Señal 15, TERM (default). Termina normalmente el proceso.
    - Señal 9, KILL. Termina inmediadamente.
    - Señal 1, HUP . Relee los archivos de configuracion
    - Man 7 signal Muestra una lista completa de las señales.

Adminstración de procesos

- Envío de señales a los procesos.
  - Por PID: **kill [señal] pid1 pid2 …**
  - Por nombre: **killall [señal] comando …**

Administración de procesos

- Administración interactiva
  - **Top**
    - **Es capaz de desplegar la información de procesos en tiempo real.**
    - **Permite clasificar, matar y reasignar prioridad a los procesos existentes.**

Administración de procesos

- Control de trabajos
  - Para ejecutar trabajos en background
    - Se agrega un **&** al final de la línea de comandos.
  - Temporalmente se detiene un programa en ejecución.
    - Ctrl z o se envía una señal 17 al proceso correspondiente
  - Administrar trabajos en background o suspendidos.
    - Listar nombres y números de trabajo: **jobs**
    - **Resume en el background: bg [%numtrab]**
    - **Resumen en el foreground: fg [%numtrab]**
    - **Envía una señal: kill [-señal] [numtrab]**

Administración de procesos

- Calendarizar un proceso para ejecutarse más tarde
  - Los trabajos a ejecutarse una sóla vez usan at, para trabajo recurrentes se aplica crontab.
  - At se utiliza de la siguiente manera:
    - at time Calendariza una tarea
    - at -l Lista las tareas calendarizadas
    - at -d Elimina un trabajo calendarizado
  - La salida no redirigida se envía por correo al usuario.
  - Root puede calendarizar tareas de los usuarios.

Configuración del bash shell

- Al terminar esta unidad debes ser capaz de:
  - Conocer como utilizar las variables locales y de ambiente.
  - Conocer como crear alias de comandos
  - Entender como el shell analiza una línea de comandos.
  - Saber como configurar archivos de arranque.

Configuración del bash shell

- Variables bash
  - Son valores nombrados.
  - Se utilizan para almacenar datos o salidas de comandos.
  - Se asignan con VARIABLE=VALOR
  - Se referencían con $VARIABLE
- Ejemplo:
  - $ NOMBRE=”DALIA CORTES”
  - $ echo “Hola $NOMBRE”

Configuración del bash shell

- Por default todas las variables son locales a un simple shell.
- Las variables de ambiente son heredas a por los shells hijos.
- Una variable de ambiente se crea con export VARIABLE=VALOR

Configuración del bash shell

- Algunas variables comunes
  - Variables de configuración
    - PS1: Controla la apariencia del prompt bash.
    - PATH: Directorios sonde se buscan los ejecutables.
    - EDITOR: Editor de textos por default
  - Variables de información
    - HOME: El directorio home del usuario
    - PWD :Tiene el nombre del directorio actual

Configuración del bash shell

- Los alias permiten crear atajos a comandos.
  - Alias dir = 'ls -la'
- Utiliza **alias** por si sólo para listar los alias existentes.

$ alias

- Utiliza **alias** seguido por un nombre de alias para ver el valor del alias.

Configuración del bash shell

- Como evitar la expansión
  - La **** quita el significado especial al siguiente caracter.
  - Por ejemplo: el signo de $, sirve para referenciar a una variable, y devolver el valor asociado.
  - Pero si no queremos darle esa función se antecede con ****.
  - Ejemplo: echo “El descuento a aplicar es de \$20.00”
  - El entrecomillado con (**'**), evita todo tipo de expansión.
  - El entrecomillado con (“) también evita la expansión, excepto para los caracteres (**`, $, \,!**).

Configuración del bash shell

- Tareas de arranque del bash: profile
  - Se localiza en /etc/profile (de alcance global), y puede crearse ~/.profile, por cada usuario.
  - Se ejecuta para intérpretes de login
  - Y por lo general se utiliza para configurar variables de ambientes y para ejecutar comandos.

Clientes de Red

- Al terminar esta unidad, debes ser capaz de:
  - Transferir archivos entre sistemas.
  - Accesar a un sistema remotamente.
  - Utilizar herramientas de diagnóstico de red.

Clientes de red

- Descarga de archivos vía http o ftp: **wget**
  - No es iteractiva, útilizable en shell scripts.
  - Puede seguir y cruzar arboles de directorios en un servidor remoto. Puede accesar a recursos web y ftp.
  - wget [-opciones] <url>
  - Ejemplo:
- wget -c http://uagrospot.blogspot.com/x/fundamentosLinux.pdf

Clientes de red

- Transferencia de archivos segura: **scp**
  - Es un remplazo seguro de rcp.
  - Es parte de ssh
  - **scp** *origen destino*
  - Los archivos remotos pueden ser especificados utilizando :
    - [usuario]@hostname:/ruta_al_archivo

Clientes de red

- Shell remoto seguro: **OpenSSH**
  - Reemplaza a las herramientas inseguras tales como telnet.
  - Permite el acceso autenticado y cifrado a sistemas remotos.
  - Ssh [username@] hostname
  - Ssh [username@] hostname comando

Herramientas de configuración básica del Sistema

- Objetivo: Al terminar este tema, debes ser capaz de:
  - Configurar la red y fecha del sistema

Herramientas de configuración básica del Sistema

- Configuración de red TCP/IP
  - Configuraciónes de red importantes
    - Configuración IP
    - Activación de dispositivo
    - Configuración de DNS
    - Puerta de enlace por default

Herramientas de configuración básica del Sistema

- Administración de conexiones ethernet
  - Las interfaces de red se nombran secuencialmente como eth0, eth1, ..., o bien con alguna contracción de la marca, y numero secuencial.
  - Se puede asignar más de una dirección a una interface, mediante el uso alias.
  - Los alias se etiqueta como eth0:1, eth0:2, etc
  - Los alias se tratan como interfaces separadas.
  - Para ver configuración de una interface utiliza: ifconfig [nombre_interface]
  - Para activar una interface: ifconfig [eth0]
  - Para desactivar una interface: ifconfig [eth0]

Herramientas de configuración básica del Sistema

- Configuración de puerta de enlace y dns.
  - La información de dns se guarda en /etc/resolv.conf
  - La direccion IP y puerta de enlace se almacena en un archivo de configuración de interface, y depende la distribución.
  - Para cambiar la puerta de enlace, utiliza

route add -net 0.0.0.0 netmask 0.0.0.0 gw <ip_gateway>

  - Para quitar una puerta de enlace, utiliza

route del -net 0.0.0.0 netmask 0.0.0.0 gw <ip_gateway>

Contacto:

M.C. Felíx Molina Ángel

Universidas Autonoma de Guerrero

molina@uagro.mx